US-PAT-NO: 5694591

DOCUMENT-IDENTIFIER: US 5694591 A

TITLE: Reducing query response time using tree balancing

DATE-ISSUED: December 2, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Du; Weimin	San Jose	CA	N/A	N/A
Shan; Ming-Chien	Saratoga	CA	N/A	N/A
Dayal; Umeshwar	Saratoga	CA	N/A	N/A

US-CL-CURRENT: 707/2, 707/10 , 707/4

ABSTRACT:

A method for optimizing data retrieval from a multidatabase system by restructuring a database query tree to optimize query response time in a two step optimization process. First, the query tree is transformed into a left deep join tree having a root query, a plurality of subordinate (descendant) query nodes and a plurality of table nodes, each subordinate query node having a left child subtree and a right child subtree. This transformation is usually the result of a first optimization scheme such as System-R. A response time for the root query and for each of the plurality of subordinate query nodes is estimated and access response times to each table node and subtree are estimated. Then, this data is utilized in the balancing of the left deep join query tree so that the cost for access to each left child subtree is substantially equal to the cost for the right child subtree. This balancing step encompasses the second phase of the query tree optimization process and includes using transformation processes such top-down, bottom-up, and a hybrid of the first two. Finally, the query is executed in a relational database to retrieve data responsive to the query in accordance with an execution plan operating according to the balanced query tree.

11 Claims, 41 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 20

---------- KWIC ---------

Brief Summary Text - BSTX (11):

The data access strategy chosen by a system optimizer specifies the exact way to structure a query (or access path) for obtaining and processing the data pages of a table. These data pages are usually organized within a relational DBMS in indexes whereby each index entry contains a key value and an identifier or pointer to the one or more rows of the table that contain the key value. Indexes are in turn stored on index pages in query tree form where there is a root page, intermediate pages that are dependent on the root page, and index leaf pages at the lowest level of the tree. Referring to the query tree shown in FIG. 2, the root and dependent nodes may be chosen so as to equalize (as much as possible) the frequencies with which a search will end in the left and

right subtrees. The frequencies are often directly proportional to the size of the data table. The resulting tree structure is called a "balanced bushy tree" and serves as a useful organizational tool for query access optimization.